Linearly bounded infinite graphs

Arnaud Carayol, Antoine Meyer

Acta Informatica manuscript No.
(will be inserted by the editor)

Arnaud Carayol · Antoine Meyer

Linearly Bounded Infinite Graphs

Received: date / Revised: date

Abstract Linearly bounded Turing machines have been mainly studied as acceptors for context-sensitive languages. We define a natural class of infinite automata representing their observable computational behavior, called linearly bounded graphs. These automata naturally accept the same languages as the linearly bounded machines defining them. We present some of their structural properties as well as alternative characterizations in terms of rewriting systems and context-sensitive transductions. Finally, we compare these graphs to rational graphs, which are another class of automata accepting the context-sensitive languages, and prove that in the bounded-degree case, rational graphs are a strict sub-class of linearly bounded graphs.

1 Introduction

One of the cornerstones of formal language theory is the hierarchy of languages introduced by Chomsky in [9]. It rests on the definition of four increasingly restricted classes of grammars, which respectively generate the *recursively enumerable*, *context-sensitive*, *context-free* and *rational* languages. All these classes were extensively studied, and have been given several alternative characterizations using different kinds of formalisms (or *acceptors*). For instance, pushdown systems characterize context-free languages, and linearly

A preliminary version of this article appeared in MFCS 2005.

Arnaud Carayol
Irisa, Campus de Beaulieu, 35042 Rennes Cedex, France
Tel.: +33299847278, Fax: +33299847171
E-mail: Arnaud.Carayol@irisa.fr

Antoine Meyer
Liafa, Université Denis Diderot, Case 7014, 2 place Jussieu, 75251 Paris Cedex 05, France
E-mail: Antoine.Meyer@liafa.jussieu.fr

bounded Turing machines (LBMs) characterize context-sensitive languages. More recently, several authors have related these four classes of languages to classes of infinite graphs (for a survey, see for instance [27]). Given a fixed initial vertex and a set of final vertices, one can associate a language to a graph by considering the set of all words labeling a path between the initial vertex and one of the final vertices. In [8], a summary of four classes of graphs accepting the four classes of the Chomsky hierarchy is presented. They are the Turing graphs [6], rational graphs [20,21,24], prefix-recognizable graphs [4,5], and finite graphs.

Several approaches exist to define classes of infinite graphs, among which we will cite three. The first one is to consider the finite acceptor of a formal language, and to build a graph representing the structure of its computations: vertices represent configurations, and each edge reflects the observable effect of an input on the configuration. One speaks of the *transition graph* of the acceptor. An interesting consequence is that the language of the graph can be deduced from the language of the acceptor it was built from. A second method proposed in [8] is to consider the Cayley-type graphs of some classes of word rewriting systems. Each vertex is a normal form for a given rewriting system, and an edge between two vertices represents the addition of a letter and re-normalization by the rewriting system. Finally, a third possibility is to directly define the edge relations in a graph using automata or other formalisms. One may speak of derivation, transduction or computation graphs. In this approach, a path no longer represents a run of an acceptor, but rather a composition of binary relations.

Both prefix-recognizable graphs and Turing graphs have alternative definitions along all three approaches. Prefix-recognizable graphs are defined as the graphs of recognizable prefix relations. In [25], Stirling characterized them as the transition graphs of pushdown systems. It was also proved that they coincide with the Cayley-type graphs of prefix rewriting systems. As for Turing graphs, Caucal showed that they can be seen indifferently as the transition and computation graphs of Turing machines [6]. They are also the Cayley-type graphs of unrestricted rewriting systems. Rational graphs, however, are only defined as transduction graphs (using rational transducers) and as the Cayley-type graphs of left-overlapping rewriting systems, and lack a characterization as transition graphs. In this paper, we are interested in defining a suitable notion of transition graphs of linearly bounded Turing machines, and to determine some of their structural properties as well as to compare them with rational graphs.

As in [6] for Turing machines, we first define a labeled version of LBMs, called LLBMs. Their transition rules are labeled either by a symbol from the input alphabet or by a special symbol denoting an internal, unobservable transition. Following an idea from [25], we consider that in every configuration of a LLBM, either internal actions or inputs are allowed, but not both at a time. This way, we can distinguish between internal and external configurations. The transition graph of a LLBM is the graph whose vertices are external configurations, and whose edges represent an input followed by any finite number of silent transitions. This definition is purely structural and associates a unique graph to any given LLBM. For convenience, we call

such graphs *linearly bounded graphs*. A similar work was proposed in [16,23], where the class of configuration graphs of LBMs up to weak bisimulation is studied. However, it provides no formal definition associating LBMs to a class of *real-time* graphs (without edges labeled by silent transitions) representing their observable computations.

To further illustrate the suitability of our notion, we provide two alternative definitions of linearly bounded graphs. First, we prove that they are isomorphic to the Cayley-type graphs of length-decreasing rewriting systems. The second alternative definition directly represents the edge relations of a linearly bounded graph as a certain kind of context-sensitive transductions. This allows us to straightforwardly deduce structural properties of linearly bounded graphs, like their closure under synchronized product (which was already known from [16]) and under restriction to a context-sensitive set of vertices. To conclude this study, we show that linearly bounded graphs and rational graphs form incomparable classes, even in the finite degree case. However, bounded degree rational graphs are a strict sub-class of linearly bounded graphs.

2 Preliminary Definitions

A labeled, directed and simple *graph* is a set $G \subseteq V \times \Sigma \times V$ where Σ is a finite set of labels and V a countable set of *vertices*. An element (s, a, t) of G is an *edge* of *source* s, *target* t and *label* a, and is written $s \xrightarrow[G]{a} t$ or simply $s \xrightarrow{a} t$ if G is understood. The set of all sources and targets of a graph is its *support* V_G. Two graph $G, H \subseteq V \times \Sigma \times V$ are isomorphic if there exists a bijection ρ from V_H to V_G such that for all $x, y \in V_H$, $x \xrightarrow[H]{a} y$ iff $\rho(x) \xrightarrow[G]{a} \rho(y)$.

A sequence of edges $s_1 \xrightarrow{a_1} t_1, \ldots, s_k \xrightarrow{a_k} t_k$ with $\forall i \in [2,k],\ s_i = t_{i-1}$ is a *path*. It is written $s_1 \xrightarrow{u} t_k$, where $u = a_1 \ldots a_k$ is the corresponding *path label*. A graph is *deterministic* if it contains no pair of edges with the same source and label. One can relate a graph to a language by considering its path language, defined as the set of all words labeling a path between two given sets of vertices.

Definition 1 The (path) language of a graph G between two sets of vertices I and F is the set

$$L(G, I, F) \ = \ \{\ w \mid s \xrightarrow[G]{w} t,\ s \in I,\ t \in F\}.$$

2.1 Linearly bounded Turing machines

We recall the definitions of context-sensitive languages and linearly bounded Turing machines. A context-sensitive language is a set of words generated by a grammar whose production rules are of the form $\alpha \to \beta$ with $|\beta| \geq |\alpha|$. Such grammars are called *context-sensitive* or sometimes *growing*. A

more operational definition of context-sensitive languages is as the class of languages accepted by *linearly bounded Turing machines* (LBMs).

Definition 2 A linearly bounded Turing machine is a tuple $M = (\Gamma, \Sigma, [,], Q, q_0, F, \delta)$, where

- Γ is a finite set of *tape symbols*,
- $\Sigma \subseteq \Gamma$ is the *input alphabet* which does not contain the symbol ε,
- $[$ and $] \notin \Gamma$ are the *left* and *right end-marker*,
- Q is a finite set (disjoint from Γ) of *control states*,
- $q_0 \in Q$ is the unique *initial state*,
- $F \subseteq Q$ is a set of *final states*,
- δ is a finite set of *transition rules* of one of the forms:

$$pA \longrightarrow qB\pm \qquad\qquad p[\longrightarrow q[+ \qquad\qquad p] \longrightarrow q]-$$

with $p, q \in Q$, $A, B \in \Gamma$ and $\pm \in \{+, -\}$.

As usual in the syntax of Turing machines, symbols $+$ and $-$ in transition rules respectively denote a move of the read head to the right and to the left. The set of configurations C_M of M is the set of words uqv such that $q \in Q$, $v \neq \varepsilon$ and $uv \in [\Gamma^*]$. The transition relation $\underset{M}{\longrightarrow}$ is a subset of $C_M \times C_M$ defined as:

$$\begin{aligned}\underset{M}{\longrightarrow} = &\{\ (upAv, uBqv) \mid pA \longrightarrow qB+ \in \delta\ \} \\ &\cup \{\ (uCpAv, uqCBv) \mid pA \longrightarrow qB- \in \delta\}\end{aligned}$$

We will simply write $\longrightarrow$ when M is understood. For any input word $w \in \Sigma^*$, the unique initial configuration is $[q_0 w]$ and a final configuration c_f is a configuration containing a terminal control state. A word w is accepted by M if $[q_0 w] \longrightarrow c_f$ where c_f is a final configuration. Quite naturally, M is *deterministic* if, from any configuration, at most one rule can be applied. Formally, for all configurations c, c_1 and c_2 such that $c \longrightarrow c_1$ and $c \longrightarrow c_2$, then $c_1 = c_2$.

Other definitions of linearly bounded machines do not use the border symbols to constrain the head inside a portion of the tape whose size equals the size of the input. Instead, they externally require that the tape size used at any point during any computation be at most k times the size of the input, where k is a fixed constant. It is a well-known fact that k can be considered equal to 1 without loss of generality, and that these definitions are equivalent to the one given above. An interesting open problem raised by Kuroda [17] concerns deterministic context-sensitive languages, which are the languages accepted by deterministic LBMs. It is not known whether they coincide with non-deterministic context-sensitive languages, as is the case for recursively enumerable or rational languages.

Note that, contrary to unbounded Turing machines, it is sufficient to only consider linearly bounded machines which always terminate, also called *terminating* machines. This is expressed by the following proposition:

Proposition 1 *For every linearly bounded Turing machine, there exists a terminating linearly bounded Turing machine recognizing the same language.*

Proof It suffices to show that, for every linearly bounded machine M whose configuration graph contains a cycle, there is an equivalent terminating machine M', i.e. a machine which always terminates and accepts the same language as M. The total number of distinct configurations of M during a run on a word of size n is bounded by k^n, where k is a constant depending on the size of the control state set and work alphabet of M. It is easy to see that a word w accepted by M must be accepted by at least one run of size less than $k^{|w|}$. Indeed, if the smallest run accepting w was longer than this bound, it would necessarily contain two occurrences of the exact same configuration, i.e. a cycle. By removing this cycle, one would obtain a shorter accepting run. Let M' be the machine which simulates M while incrementing a n-digit counter in base k encoded in the alphabet and stops the run if the counter overflows. By construction, M' accepts the same language as M and is terminating. □

Similar arguments are used in [17] to show that the set of words on which a linearly bounded machine has an infinite run is context-sensitive. Another property of context-sensitive languages is that they are closed under union and complement.

Theorem 1 ([14,26]) *The context-sensitive languages over a finite alphabet Γ form an effective Boolean algebra.*

The notion of space-bounded Turing machine can be extended to any bound $f : \mathbb{N} \mapsto \mathbb{N}$ such that for all $n \geq 0$, $f(n) \geq \log(n)$. The set of all languages accepted by a Turing machine working in space $f(n)$ on an entry of size n is written NSPACE$[f(n)]$. The following theorem states that the space hierarchy is strict.

Theorem 2 ([13,14]) *For all space-constructible f and g in $\mathbb{N} \mapsto \mathbb{N}$ such that $\lim_{n\to+\infty} \frac{f(n)}{g(n)} = 0$, we have* NSPACE$[f(n)] \subsetneq$ NSPACE$[g(n)]$.

In particular, the class NSPACE$[2^n]$ of languages recognizable in exponential space strictly contains the class of context-sensitive languages (or NSPACE$[n]$).

2.2 Rational graphs

Consider the product monoid $\Sigma^* \times \Sigma^*$, whose elements are pairs of words (u, v) in Σ^*, and whose composition law is defined by $(u_1, v_1) \cdot (u_2, v_2) = (u_1u_2, v_1v_2)$. A finite transducer is an automaton over $\Sigma^* \times \Sigma^*$ with labels in $(\Sigma \cup \{\varepsilon\}) \times (\Sigma \cup \{\varepsilon\})$. Transducers accept the rational subsets of $\Sigma^* \times \Sigma^*$, which are seen as binary relations on words and called rational transductions. We do not distinguish a transducer from the relation it accepts and write $(w, w') \in T$ if the pair (w, w') is accepted by T. Graphs whose vertices are words and whose edge relations are defined by transducers (one per letter in the label alphabet) are called rational graphs.

Definition 3 ([20]) A rational graph labeled by Σ with vertices in Γ^* is given by a class of transducers $(T_a)_{a\in\Sigma}$ over Γ. For all $a \in \Sigma$, $(u, a, v) \in G$ if and only if $(u, v) \in T_a$.

For $w \in \Sigma^+$ and $a \in \Sigma$, we recursively define $T_{wa} = T_w \circ T_a$, where $\circ$ denotes the standard relational composition, and we write $u \xrightarrow{w} v$ if and only if $(u, v) \in T_w$. In general, there is no bound on the size difference between input and output in a transducer (and hence between the lengths of two adjacent vertices in a rational graph). Interesting sub-classes are obtained by enforcing some form of synchronization. The most well-known was defined by Elgot and Mezei [11] (see also [12]) as follows. A transducer over Σ with initial state q_0 is (left-)synchronized if for every path $q_0 \xrightarrow{x_1/y_1} q_1 \dots q_{n-1} \xrightarrow{x_n/y_n} q_n$, there exists $k \in [0, n]$ such that for all $i \in [1, k]$, x_i and y_i belong to Σ and either $x_{k+1} \dots x_n = \varepsilon$ and $y_{k+1} \dots y_n \in \Sigma^*$ or $y_{k+1} \dots y_n = \varepsilon$ and $x_{k+1} \dots x_n \in \Sigma^*$. A rational graph defined by synchronized transducers will simply be called a synchronized (rational) graph.

Rational graphs form a class of infinite acceptors for context-sensitive languages. For a discussion on the expressive power of the sub-classes of rational graph seen as language acceptors see [3].

Theorem 3 ([21]) *The languages accepted by the rational graphs between a rational set of initial vertices and a rational set of final vertices are the context-sensitive languages.*

This result can be slightly strengthened in the case of a single initial and final vertex.

Corollary 1 *For any rational graph G labeled by Σ, pair (i, f) of vertices of G, and symbol $\sharp \notin \Sigma$, the language $L_G = \{i\sharp w\sharp f \mid w \in L(G, \{i\}, \{f\})\}$ is context-sensitive.*

Proof Let G be a rational graph labeled by Σ with vertices in Γ^* and defined by a family of transducers $(T_a)_{a\in\Sigma}$. Let $\bar{\Gamma}$ and $\tilde{\Gamma}$ be two finite alphabets disjoint from but in bijection with Γ. For any $x \in \Gamma$, we write $\bar{x}$ (resp. $\tilde{x}$) the corresponding symbol in $\bar{\Gamma}$ (resp. $\tilde{\Gamma}$). We consider the rational graph H labeled by $\Xi = \Sigma \cup \bar{\Gamma} \cup \tilde{\Gamma}$ defined by the family of transducers $(T_x)_{x\in\Xi}$ where for all $x \in \Gamma$, $T_{\bar{x}} = \{(u, ux) \mid u \in \Gamma^*\}$ and $T_{\tilde{x}} = \{(xu, u) \mid u \in \Gamma^*\}$.

By Thm. 3, the language $L = L(H, \varepsilon, \varepsilon) \cap \bar{\Gamma}^*\Sigma^*\tilde{\Gamma}^*$ is context-sensitive. By construction, L is equal to $\{\bar{i}w\tilde{f} \mid i, f \in \Gamma^* \text{ and } w \in L(G, \{i\}, \{f\})\}$. It follows that L_G is context-sensitive. □

3 Linearly Bounded Graphs

3.1 LBM Transition Graphs

Following [6], we define the notion of labeled linearly bounded Turing machine (LLBM). This notion is very close to the notion of off-line Turing machine

[13]. It is essentially equivalent to an off-line Turing machine with a one-way input tape and a two-way linearly bounded work tape. As in standard definitions of LBMs, the transition rules can only move the head of the LLBM between the two end markers [and]. In addition, a silent step can decrease the size of the configuration (without removing the markers) and a Σ-transition can increase the size of the configuration by one cell. This ensures that while reading a word of length n, the labeled LBM uses at most n cells.

Definition 4 A labeled linearly bounded Turing machine is a tuple $M = (\Gamma, \Sigma, [,], Q, q_0, F, \delta)$ as in Definition 2, where all components but δ are defined similarly, and δ is a finite set of *labeled* transition rules of one of the forms:

$$\begin{array}{lll} pA \xrightarrow{\varepsilon} qB\pm & p[\xrightarrow{\varepsilon} q[+ & p] \xrightarrow{\varepsilon} q]- \\ pB \xrightarrow{a} qAB & p] \xrightarrow{a} qA] & pA \xrightarrow{\varepsilon} q \end{array}$$

with $p, q \in Q$, $A, B \in \Gamma$, $\pm \in \{+, -\}$ and $a \in \Sigma$.

Configurations are defined similarly to the previous case. However, the transition relation is now labeled. For all $x \in \Sigma \cup \{\varepsilon\}$, the relation $\xrightarrow[M]{x}$ is a subset of $C_M \times C_M$ defined as:

$$\begin{aligned} \xrightarrow[M]{x} = {} & \{\ (upAv, uBqv) \mid pA \xrightarrow{x} qB+ \in \delta\ \} \\ & \cup \{\ (uCpAv, uqCBv) \mid pA \xrightarrow{x} qB- \in \delta\} \\ & \cup \begin{cases} \{\ (upAv, uqv) \mid pA \xrightarrow{x} q \in \delta\} & \text{with } x = \varepsilon \\ \{\ (upAv, uqBAv) \mid pA \xrightarrow{x} qBA \in \delta\} & \text{with } x \in \Sigma. \end{cases} \end{aligned}$$

Transitions are composed by defining $\xrightarrow{wx}$ as $(\xrightarrow{w} \circ \xrightarrow{x})$ for all $w \in \Sigma^*$. The definition of runs also changes to reflect the fact that input words are no longer present on the tape but are read as the run progresses. The unique initial configuration is thus $[q_0]$. A word w is accepted by M if $[q_0] \xrightarrow{w} c_f$ where c_f is a final configuration. M is *deterministic* if, from any configuration, either all possible moves are labeled by *distinct* letters of Σ, or there is only one possible move labeled by ε. Formally, for all configurations c, c_1 and c_2 such that $c \xrightarrow{x} c_1$ and $c \xrightarrow{y} c_2$, either x and y belong to Σ and if $c_1 \neq c_2$ then $x \neq y$, or $x = y = \varepsilon$ and $c_1 = c_2$.

Labeled LBMs are as expressive as classical LBMs.

Proposition 2 *A language is (deterministic) context-sensitive if and only if it is accepted by a (deterministic) labeled linearly bounded Turing machine.*

Proof Let us first consider a context-sensitive language L accepted by a terminating LBM N (by Prop. 1). We sketch the construction of a LLBM $M = (\Gamma, \Sigma, [,], Q, q_0, F, \delta)$ accepting L. Let q_A, q_R and q_S be three states in Q. In a configuration of the form $[wq_X]$ for $w \in \Sigma^*$ and $q_X \in \{q_A, q_R\}$, M reads an input letter $a \in \Sigma$ and goes to the configuration $[waq_s]$. Then it simulates N on wa using only ε-transitions while remembering wa. Using the

same work alphabet as N, this would require the use of $2|wa|$ cells. However using an increased work alphabet (or a second work tape), this can be done using only $|wa|$ cells. If N accepts (resp. rejects) wa, M restores wa on its (primary) work tape and steps in configuration $[waq_A]$ (resp. $[waq_R]$). By taking $F = \{q_A\}$ and $q_0 = q_A$ if ε belongs to L and $q_0 = q_R$ otherwise, it is easy to see that the set of words accepted by M from $[q_0]$ is precisely L. Note that non-deterministic behavior can only appear in N while simulating M. Hence, if M is deterministic then so is N.

Conversely, let M be a LLBM accepting a language $L \subseteq \Sigma^*$. We describe a LBM N accepting L with two tapes: the input tape and work tape. It is well known that this model is equivalent to LBMs with one tape as presented in Sec. 2.1 (see for example [13]). To each tape corresponds a set of control states: the set of work states Q_Γ contains the set Q of states of M and the set of input states Q_Σ is reduced to $\{q_i, q_A\}$ respectively the initial and accepting state.
The machine N working on word $w \in \Sigma^*$ starts with the configuration $([q_i w], [q_0])$. From a configuration $([w_1 q_i w_2], [uqv])$ with $w_1, w_2 \in \Sigma^*$, $q \in Q$ and $u, v \in \Gamma^*$, N can non-deterministically simulate any ε-transition of N that can be applied to the configuration represented by the work tape. The deletion rules are simulated by shifting all tape content on the right of the read head by one cell to the left. Moreover N can non-deterministically simulate any a-transition for $a \in \Sigma$ provided that the input head is on top of the symbol a in which case it is moved one cell to the right. The insertion rules are simulated by shifting the work tape content to the right of the read head by one cell on the right.
The machine M enters the accepting state q_A if the input head is on top of the right border symbol] and the work state is a final state of N. It follows from the construction of M that w is accepted by M if and only if it is accepted by N. Moreover, if N is deterministic then so is M. □

Remark 1 For convenience, one may consider LBMs whose initial configuration is not of the form $[q_0]$ but is any fixed configuration c_0. This does not add any expressive power, as can be proved by a simple encoding of c_0 into the control state set of the machine, which will not be detailed here.

Remark 2 For simplicity, the above definition forces the insertion of a new tape cell each time a letter is read. More relaxed forms where a cell deletion or rewriting can occur during an input may be considered without any consequence for the results. Similarly, rules which do not move the read head can be allowed.

Let $M = (\Gamma, \Sigma, [,], Q, q_0, F, \delta)$ be a LLBM, we define its configuration graph

$$C_M \;=\; \big\{(c, a, c') \mid c \xrightarrow[M]{a} c' \quad \text{for } a \in \Sigma \cup \{\varepsilon\}\big\}.$$

The vertices of this graph are all configurations of M, and its edges denote the transitions between them, including ε-transitions. One may wish to only consider the behavior of M from an external point of view, i.e. only looking at the sequence of inputs. This means one has to find a way to conceal ε-transitions without changing the accepted language or destroying the

structure. One speaks of the *transition graph* of an acceptor, as opposed to its configuration graph.

In [25], Stirling mentions a normal form for pushdown automata which allows him to consider a structural notion of transition graphs, without relying on the naming of vertices. We first recall this notion of *normalized* systems adapted to labeled LBMs. A labeled LBM is *normalized* if its set of control states can be partitioned in two subsets: one set of *internal* states, noted Q_ε, which can perform ε-rules and only ε-rules, and a set of *external* states noted Q_Σ, which can only perform Σ-rules. More formally:

Definition 5 A labeled LBM $M = (\Gamma, \Sigma, [,], Q, q_0, F, \delta)$ is *normalized* if there are disjoint sets Q_Σ and Q_ε such that $Q = Q_\varepsilon \cup Q_\Sigma$, $F \subseteq Q_\Sigma$, and

$$pB \xrightarrow{a} qAB \in \delta \implies p \in Q_\Sigma,$$
$$pA \xrightarrow{\varepsilon} qB\pm \in \delta \text{ or } pA \xrightarrow{\varepsilon} q \in \delta \implies p \in Q_\varepsilon,$$
$$p \in Q_\varepsilon \implies \text{for all } A \in \Gamma,\ \exists pA \xrightarrow{\varepsilon} qB\pm \in \delta.$$

This definition implies in particular that a control state from which there exists no transition must belong to Q_Σ. A configuration is external if its control state is in Q_Σ, and internal otherwise. This makes it possible to *structurally* distinguish between internal vertices, which have one or more outgoing ε-edges, and external ones which only have outgoing Σ-edges or have no outgoing edges. Given any labeled LBM, it is always possible to normalize it without changing the accepted language.

Proposition 3 *Every labeled linearly bounded machine can be normalized without changing the accepted language.*

Proof Let M be a LLBM, we know from Prop. 2 that the language L accepted by M is context-sensitive. Consider the LLBM M' accepting L obtained in the construction of the proof of Prop. 2. It is easy to see that M' is normalized: q_A and q_R are the only states that can perform transitions labeled by Σ, and they cannot perform ε-transitions. All other states can only perform ε-transitions. To verify the third condition in the definition of normalized LLBMs, it suffices to add instruction $pA \xrightarrow{\varepsilon} pA$ for every p and A for which the condition is violated. Moreover, the unique terminal state is q_A, which is external. □

Remark 3 Normalization as previously described preserves determinism.

From this point on, unless otherwise stated, we will only consider normalized LLBMs. We can now define the transition graph of a LLBM as the ε-closure of its configuration graph, followed by a restriction to its set of external configurations (which happens to be a rational set).

Definition 6 Let $M = (\Gamma, \Sigma, [,], Q, q_0, F, \delta)$ be a (normalized) LLBM, and C_Σ be its set of external configurations. The transition graph of M is

$$G_M = \left\{(c, a, c') \mid c, c' \in C_\Sigma,\ a \in \Sigma,\ \wedge\ c \xrightarrow[M]{a\varepsilon^*} c'\right\}.$$

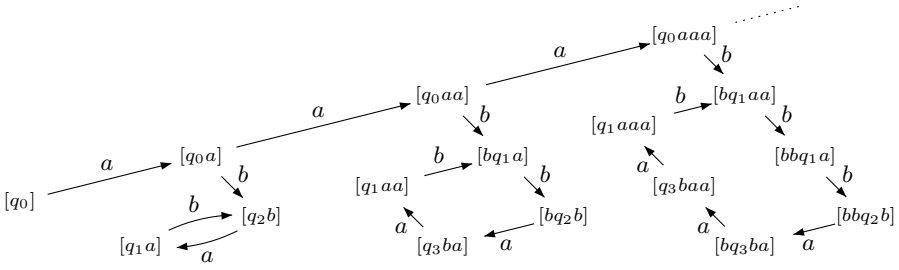

Fig. 1 The transition graph of a labeled LBM accepting $\{(a^n b^n)^+ \mid n \geq 1\}$.

We now define the class of linearly bounded graphs as the closure under isomorphism of transition graphs of labeled LBMs.

Example 1 Figure 1 shows the transition graph of the normalized LLBM M whose rules are:

$$q_0] \xrightarrow{a} q_0 a] \qquad q_1 a \xrightarrow{b} q_1 b+ \qquad q_2 b \xrightarrow{a} q_3 a- \qquad q_3 b \xrightarrow{a} q_3 a-$$
$$q_0 a \xrightarrow{a} q_0 aa \qquad q_1] \xrightarrow{\varepsilon} q_2]- \qquad\qquad q_3[\xrightarrow{\varepsilon} q_1[+$$
$$q_0 a \xrightarrow{b} q_1 b+$$

and whose unique accepting state is q_2. This machine accepts the language $\{(a^n b^n)^+ \mid n \geq 1\}$, which is also the language of paths of the graph between vertex $[q_0]$ and the set $[b^* q_2 b]$. For the sake of clarity, only the part of the graph reachable from configuration $[q_0]$ is shown. We will see in Sect. 4 that this sub-graph is still a linearly bounded graph.

3.2 Alternative definitions

This section provides two alternative definitions of linearly bounded graphs. In [8], it is shown that all previously mentioned classes of graphs can be expressed in a uniform way in terms of Cayley-type graphs of certain classes of rewriting systems. We show that it is also the case for linearly bounded graphs, which are the Cayley-type graphs of length-decreasing rewriting systems. The second alternative definition we present changes the perspective and directly defines the edges of linearly bounded graphs using incremental context-sensitive transductions. This variety of definitions will allow us to prove in a simpler way some of the properties of linearly bounded graphs.

3.2.1 Cayley-type graphs of decreasing rewriting systems

We first give the relevant definitions about rewriting systems and Cayley-type graphs. A *word rewriting system* R over alphabet Γ is a subset of $\Gamma^* \times \Gamma^*$. Each element $(l, r) \in R$ is called a *rewriting rule* and noted $l \to r$. The words

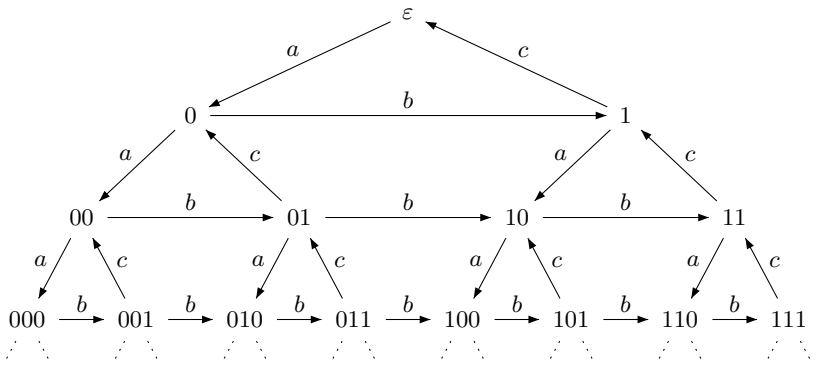

Fig. 2 Cayley-type graph of the rewriting system $R = \{a \to 0,\ b \to b,\ 0b \to 1,\ 1b \to b0,\ c \to c,\ 1c \to \varepsilon\}$, with $\Sigma = \{a, b, c\}$ and $\Gamma = \{a, b, c, 0, 1\}$. Rules $b \to b$ and $c \to c$ ensure that no normal form of R contains b or c. As an example, edge $01 \xrightarrow{b} 10$ is justified by the derivation $01b \to 0b0 \to 10$.

l and r are respectively called the left- and right-hand side of the rule. The *rewriting relation* of R is the binary relation

$$\{(ulv, urv) \mid u, v \in \Gamma^*,\ l \to r \in R\}$$

which we also denote by R, consisting of all pairs of words (w_1, w_2) such that w_2 can be obtained by replacing (*rewriting*) an instance of a left-hand side l in w_1 with the corresponding right-hand side r. The reflexive and transitive closure R^* of this relation is called the *derivation* of R. Whenever for some words u and v we have uR^*v, we say R rewrites u into v. A word which contains no left-hand side is called a *normal form*. The set of all normal forms of R is written $\mathrm{NF}(R)$.

One can associate a unique infinite graph to any rewriting system by considering its *Cayley-type graph* defined as follows:

Definition 7 The Σ-labeled Cayley-type graph of a rewriting system R over Γ, with $\Sigma \subseteq \Gamma$, is the infinite graph

$$G_R \ = \ \{(u, a, v) \mid a \in \Sigma,\ u, v \in \mathrm{NF}(R),\ uaR^*v\}.$$

The class of rewriting systems we consider is that of *finite length decreasing word rewriting systems*, i.e. rewriting systems with a finite set of rules of the form $l \to r$ with $|l| \geq |r|$, which can only preserve or decrease the length of the word to which they are applied. The reason for this choice is that the derivation relation of such a system coincides with arbitrary compositions of ε-rules of a given labeled LBM.

Example 2 Figure 2 shows the Cayley-type graph of a simple decreasing rewriting system.

3.2.2 Incremental context-sensitive transduction graphs

The notion of *computation graph* was first systematically used in [8] (where the terminology used is *relation graph*). It corresponds to the graphs defined by the *transductions* (i.e. binary relations on words) associated to a class of finite machines. These works prove that for pushdown automata and Turing machines, the classes of transition and computation graphs coincide. We show that it is also the possible to give a definition of linearly bounded graphs as the computation graphs of a certain class of LBMs, or equivalently as the graphs defined by a certain class of context-sensitive transductions.

A relation R is recognized by a LBM M if the language $\{u\#v \mid (u,v) \in R\}$ where $\#$ is a fresh symbol is accepted by M. However, this type of transductions generates more than linearly bounded graphs. Even if we only consider linear relations (i.e relations R such that there exists c and $k \in \mathbb{N}$ such that $(u,v) \in R$ implies $|v| \leq c \cdot |u| + k$), we obtain graphs accepting the languages recognizable in exponential space (NSPACE[2^n]) which strictly contain the context-sensitive languages (cf Thm. 2). We need to consider relations for which the length difference between a word and its image is bounded by a certain constant. Such relations can be associated to LBMs as follows:

Definition 8 A *k-incremental context-sensitive transduction* T over Γ is defined by a LBM recognizing a language $L \subseteq \{u\#v \mid u,v \in \Gamma^* \text{ and } |v| \leq |u| + k\}$ where $\#$ does not belong to Γ. Relation T is defined as $\{(u,v) \mid u\#v \in L\}$.

The synchronized relations of finite image (i.e for u there are finitely many v such that $(u,v) \in R$) provide a first example of k-incremental context-sensitive transductions.

Proposition 4 *For any synchronized relation R of finite image, there exists a constant $k \in \mathbb{N}$ such that R is a k-incremental context-sensitive transduction.*

Proof Let $R \subseteq \Gamma^* \times \Gamma^*$ be a synchronized relation of finite image. It follows from the definition of synchronized relations that R is equal to a finite union:

$$\left(\bigcup_{i\in I} S_i \cdot (A_i \times \varepsilon)\right) \cup \left(\bigcup_{j\in J} S_j \cdot (\varepsilon \times B_j)\right)$$

where for all $k \in I \cup J$, S_k is a length-preserving relation[1] and for all $i \in I$ and $j \in J$, A_i and B_j are rational subsets of Γ^*.
As R has a finite image, for all $j \in J$ the set B_j is necessarily finite. Let k be the maximal length of a word in $\bigcup_{j\in J} B_j$, it is easy to check that for all pairs of words $(u,v) \in R$, $|v| \leq |u|+k$. Moreover the language $\{u\#v \mid (u,v) \in R\}$ is context-sensitive. Hence R is a k-incremental context-sensitive transduction. □

The following proposition states that incremental context-sensitive transductions of a given level form a Boolean algebra.

[1] A synchronized relation R is length-preserving if $\forall (x,y) \in R, |x| = |y|$

Proposition 5 *For all k-incremental context-sensitive transductions T and T' over Γ^*, $T \cup T'$, $T \cap T'$ and $\overline{T} = E_k - T$ (where E_k is $\{(u,v) \mid 0 \leq |v| \leq |u| + k\}$) are incremental context-sensitive transductions.*

Proof The closure under union follows from that of context-sensitive languages. The proof of closure under complement is a straightforward consequence of the closure under complement of context-sensitive languages (cf Thm. 1). Let $T \subset \Gamma^* \times \Gamma^*$ be a k-incremental context-sensitive transduction. By definition, the set $L = \{u\#v \mid (u,v) \in T\}$ is context-sensitive. It is straightforward to check that the set $L' = \{u\#v \mid (u,v) \in E_k - T\}$ is equal to:

$$\overline{L} \cap \{u\#v \mid |u| \leq |v| + k\}.$$

As the context-sensitive languages are closed under complement and intersection, L' is context-sensitive. Hence, $\bar{T}$ is a k-incremental context-sensitive transduction. □

The canonical graph associated to a finite set of transductions is called a *transduction graph*. Relating graphs to a class of binary relations on words was already used to define rational graphs and their sub-classes.

Definition 9 The Σ-labeled transduction graph of a finite set of incremental context-sensitive transductions $(T_a)_{a \in \Sigma}$ is

$$G_T \;=\; \{(u,a,v) \mid a \in \Sigma \text{ and } (u,v) \text{ is recognized by } T_a\}.$$

Example 3 The linearly bounded graph of Ex. 1 is isomorphic to the transduction graph of the following set of incremental context-sensitive transductions:

$$T_a = \{(\#a^n, \#a^{n+1}) \mid n \geq 0\} \cup \{(b^m a^n, b^{m-1} a^{n+1}) \mid m \geq 1, n \geq 0\},$$
$$T_b = \{(\#a^n, a^{n-1}b) \mid n \geq 1\} \cup \{(a^m b^n, a^{m-1} b^{n+1}) \mid m \geq 1, n \geq 0\}.$$

The symbol # is needed to distinguish a vertex directly reachable through a sequence of a's from a vertex reachable through a sequence of the form $(a^n b^n)^*$.

As for the Cayley-type graph of Ex. 2, it can be seen as the transduction graph of the set of incremental context-sensitive transductions $\{T_a, T_b, T_c\}$, where T_a adds a 0 and T_c removes a 1 to the right of a binary number, and T_b implements binary increment.

Length-preserving context-sensitive transductions have already been extensively studied in [18]. In the rest of this presentation, unless otherwise stated, we will only consider 1-incremental transductions without loss of generality regarding the obtained class of graphs: indeed, any k-incremental transduction graph is isomorphic to a 1-incremental one over an increased alphabet.

3.2.3 Equivalence of all definitions

We now prove that both classes of Cayley-type graphs of decreasing rewriting systems, and incremental context-sensitive transduction graphs define precisely the class of linearly bounded graphs, up to isomorphism (i.e. up to vertex renaming).

Theorem 4 *For any graph G, the following statements are equivalent:*

1. *G is isomorphic to the transitions graph of a labeled LBM,*
2. *G is isomorphic to the Cayley-type graph of a finite length-decreasing system,*
3. *G is isomorphic to a context-sensitive transduction graph.*

Proof 1 $\implies$ 2: Let $M = (\Gamma, \Sigma, [,], Q, q_0, F, \delta)$ be a normalized labeled linearly bounded Turing machine, with $\Sigma \cap \Gamma = \emptyset$. As M is normalized, its control states can be partitioned into Q_Σ and Q_ε (see Definition 5). Let $\Gamma' = \Gamma \cup \{[,]\}$. We build a finite length-decreasing rewriting system R whose Cayley-type graph is the transition graph of M. Let the alphabet of R be $\Delta = \Sigma \cup \Gamma' \cup (\Gamma' \times Q) \cup S$, where $S = \{v_a, v'_a, s_a \mid a \in \Sigma\}$ is a new set of symbols disjoint from Γ and Q. Elements (x, q) of $\Gamma' \times Q$ will be noted x_q. For convenience, for any set X, $X_\bullet$ will denote $X \cup (X \times Q_\Sigma)$, and $x_\bullet$ any symbol in $\{x\}_\bullet$.

There are several important points which the rules of R must ensure:

1. Only words of the form $(\varepsilon \cup [_\bullet)\, \Gamma_\bullet^*\, (\varepsilon \cup]_\bullet)$, i.e. words in which only external control states may occur and in which no symbol occurs to the left of a left bracket or to the right of a right bracket, should be normal forms (please be aware that $\Gamma_\bullet$ denotes the set $\Gamma \cup (\Gamma \times Q_\Sigma)$, and not $\Gamma \cup (\Gamma \times Q)$):
$$x[_\bullet \to [_\bullet, \qquad]_\bullet y \to]_\bullet, \qquad s \to s$$
for all $x \in \Delta,\ y \in \Delta \setminus \Sigma,\ s \in \Sigma \cup S \cup (\Gamma' \times Q_\varepsilon)$.
2. When a letter a in Σ is added to the right of an irreducible word u, one should ensure that u actually represents a legal configuration[2], i.e. that u is of one of the forms $[_q\Gamma^*]$, $[\Gamma^* A_q \Gamma^*]$ or $[\Gamma^*]_q$ for $A \in \Gamma$, $q \in Q$:
$$]a \to v_a], \qquad]_q a \to v_a{}']_q, \qquad Av_a \to v_a A, \qquad A_q v_a \to v_a{}' A_q,$$
$$Av_a{}' \to v_a{}' A, \qquad [_q v_a \to [s_a, \qquad [v_a{}' \to [s_a.$$
3. Finally, once it has been made sure that the word represents a legitimate LBM configuration, one should simulate an insertion operation followed by any number of ε-transitions of the LBM:
$$s_a A \to A s_a, \qquad s_a B_p \to C_q B$$
for all $a \in \Sigma,\ A, B, C \in \Gamma$ and $pB \xrightarrow{a} qCB \in \delta$, and
$$A_p C \to B C_q, \qquad C A_p \to C_q B \qquad A_p C \to C_q$$
for all $pA \xrightarrow{\varepsilon} qB+,\ pA \xrightarrow{\varepsilon} qB-,\ pA \xrightarrow{\varepsilon} q \in \delta$ (respectively).

[2] Note that this encoding of configuration differs from the one used previously, where control states were standalone symbols.

There is an edge $u \xrightarrow{a} v$ in the Cayley-type graph G_R of R if and only if u and v are words representing valid configurations of M from which no ε-transition can be performed, i.e. observable configurations, and there exists a sequence of transitions labeled by $a\varepsilon^*$ of M by which u reaches v. There is a bijection between the edges and vertices of G_R and the transition graph of M, hence these two graphs are isomorphic.

$2 \implies 3$: Let R be a finite length-decreasing rewriting system, G_R its Cayley-type graph. For each letter a, we will show that the relation

$$T_a = \{(ua, v) \mid u \xrightarrow[G_R]{a} v\} = \{(ua, v) \mid u, v \in \mathrm{NF}(R) \wedge uaR^*v\}$$

is an incremental context-sensitive transduction by building a LBM $M_a = (\Gamma, \Sigma, [,], Q, q_0, F, \delta)$ recognizing T_a.

For every pair (ua, v), M_a starts in configuration $ua\#v$, and first has to check that u is a normal form of R by verifying that it contains no left-hand side of any rule in R. Second, M_a simulates the derivation of R on ua, applying one rewriting rule at a time until a normal form is reached. Due to non-determinism, there might be unsuccessful runs, but the pair is accepted if and only if one run reaches configuration $v\#v$, meaning that R can normalize ua into v. Hence, a pair (ua, v) is in T_a if and only if $(u, a, v) \in G_R$, meaning that the transduction graph of $(T_a)_{a\in\Sigma}$ is isomorphic to G_R.

$3 \implies 1$: Let $T = (T_a)_{a\in\Sigma}$ be a finite set of incremental context-sensitive transductions defining a graph G_T, each T_a being recognized by an LBM M_a. We informally describe a normalized labeled LBM M whose transition graph G_M is isomorphic to G_T.

Let q be the unique external control state of M, M should have a run labeled by $a\varepsilon^*$ between configurations qu and qv whenever $(u, v) \in T_a$, or equivalently whenever the word $u\#v$ is accepted by M_a. This is done as follows. First, starting from configuration qu, M should perform an a-labeled transition, increasing its available tape space by 1, and step into an internal control state. It should then guess a word v, and write $u\#v$ on its tape. Since T_a is incremental, this can be done using no more than $|u| + 1$ tape cells by writing two symbols in each cell. Then, M simulates the LBM M_a on input word $u\#v$, while keeping an intact copy of v on the tape (this can again be done by a simple alphabet encoding). If the simulated run of M_a succeeds, M steps into external configuration qv by restoring the saved copy of v on the tape, otherwise it loops in a non-accepting internal state. By this construction, there is an edge (qu, a, qv) in G_M if and only if there is an edge (u, a, v) in G_T, hence both graphs are isomorphic. □

This shows that the three types of graphs presented in this section actually all define the same class, namely that of linearly bounded graphs. This variety of definitions will allow us to prove in a simpler way some of the properties of linearly bounded graphs.

4 Structural properties

Now that the class of linearly bounded graphs has been defined using three different formalisms, we can easily deduce some of their structural properties.

In particular, we look at the languages accepted by linearly bounded graphs and some of their closure properties. We also give some insight about the relation between linearly bounded graphs and deterministic context-sensitive languages, and conclude with a few logical properties. But first, we compare our notions to related work.

4.1 Comparison with existing work

We now give a precise comparison of linearly bounded graphs with the restriction of Turing graphs ([6]) to the linearly bounded case, and with the configuration graphs considered in [16].

4.1.1 Configuration graphs

In [16], the configuration graphs of a class of offline linearly bounded machines very similar to our labeled LBM are considered. However, they include in their definition a restriction to the set of configurations reachable from the initial configuration. Therefore, their class of configuration graphs is incomparable to ours. As we will see in Prop. 8, the linearly bounded graphs are closed under restriction to the set of vertices reachable from a given vertex. Hence, it is not necessary to impose this restriction directly in the definition of the configuration graph.

Apart from this restriction, our class of configuration graphs coincides with the configuration graphs of [16] and with those of [6] in the case of linearly bounded Turing machines.

4.1.2 Transition graphs

Knapik and Payet do not consider transition graphs: instead of characterizing the ε-closure of configuration graphs, they prove a closure property of this class up to weak bisimulation [19], which is not a structural characterization. Nevertheless, it is straightforward to prove that their results can be extended to the class of transition graphs considered up to isomorphism, as mentioned in Sect. 4.

In [6], Caucal defines the transition graphs of Turing machines from their configuration graphs using the very general notion of ε-closure: wherever there is a path labeled by $\varepsilon^* a \varepsilon^*$ in the configuration graph, there is an edge labeled by a in its ε-closure, for every letter a. Furthermore, the definition allows the restriction to an arbitrary rational set of vertices (potentially the whole set of vertices). Figure 3 illustrates on a small configuration graph the difference between these two approaches.

Our approach has two main advantages. First, the ε-closure operation as defined by Caucal may give rise to spurious non-determinism in the obtained graphs. This additional complexity is artificial and is eliminated using our definition. Furthermore, our notion is purely structural, and does not rely on the naming of vertices. But interestingly, we can show that both classes still coincide up to isomorphism: for every labeled linearly bounded machine

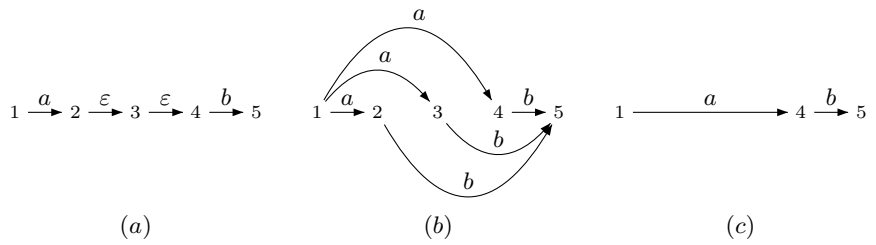

Fig. 3 (a) Normalized LLBM configuration graph. (b) Transition graph of (a) as defined in [6] (with no restriction on vertices). (c) Linearly bounded graph associated to (a).

M and rational set R, one can define a machine M' such that the transition graph of M following Caucal's definition with respect to R is isomorphic to the linearly bounded graph corresponding to M' in our framework.

Proposition 6 *The class of linearly bounded graphs coincides, up to isomorphism, with the restriction of Turing graphs to the (uniform) linearly bounded case.*

Proof Let M be a normalized LLBM, C its configuration graph, R its (rational) set of external configurations and G its transition graph defined as

$$G = \{c \xrightarrow{a} c' \mid c, c' \in R \wedge c \xrightarrow{a\varepsilon^*} c' \in C\}.$$

Consider the graph G' defined as the restriction to R of the ε-closure of C.

$$G' = \{c \xrightarrow{a} c' \mid c, c' \in R \wedge c \xrightarrow{\varepsilon^* a \varepsilon^*} c' \in C\}.$$

The graphs G and G' are equal, since by definition of external vertices, they have no outgoing ε-transitions.

Conversely, let M be a (not necessarily normalized) LLBM, C its configuration graph, and R a rational set of configurations of M. The binary relation $\xrightarrow{\varepsilon^* a \varepsilon^*}$ in $C \times C$ is a 1-incremental context-sensitive transduction. Hence the graph G defined as the restriction to R (see Prop. 8) of the ε-closure of C is a linearly bounded graph. □

4.2 Languages

It is quite obvious that the language of the transition graph of a LLBM M between the vertex representing its initial configuration and the set of vertices representing its final configurations is the language of M. In fact, the choice of initial and final vertices has no importance in terms of the class of languages one obtains.

Proposition 7 *The languages of linearly bounded graphs between an initial vertex i and a finite set F of final vertices are the context-sensitive languages.*

Proof Let G be a linearly bounded graph labeled by Σ defined by a family $(T_a)_{a\in\Sigma}$ of 1-incremental context-sensitive transductions. We will prove that $L(G,i,F)$ is context-sensitive even if F is a context-sensitive set.

Let $\# \notin \Sigma$ be a new symbol, we consider the graph $\overline{G}$ obtained from G by adding a loop labeled by $\#$ on each vertex in F. Obviously, $\overline{G}$ is a linearly bounded graph as $\xrightarrow{\#} = \{(f,f) \mid f \in F\}$ is a 0-incremental context-sensitive transduction. By Thm. 4, G is the transition graph of a LLBM M. Let c_0 be the configuration of M corresponding to the vertex i in G. Consider the LLBM M' whose initial configuration is i (see Rem. 1) and whose set of final states are the states that appear in the left-hand side of a $\#$-rule (without loss of generality, we can assume that the $\#$-rules do not depend on the current value of the cell). It is easy check that M' accepts $L(G,i,F)$ and by Prop. 2, $L(G,i,F)$ is context-sensitive.

For the converse implication, let L be a context-sensitive language and M the normalized LLBM constructed in Prop. 2 accepting L. The transition graph of M traces L from the initial configuration to an infinite set of configurations (i.e the configurations with state q_A). Therefore, we need to adapt the construction of Prop. 2. We add a new state q_f to the machine M and whenever M makes an ε-transition to enter state q_A, we add the ability to enter configuration $[q_f]$ by a sequence of ε-transitions. As $[q_f]$ has no out-going edges, it is an external configuration and it is easy to see that $L = L(G, c_0, [q_f])$. □

Remark 4 When a linearly bounded graph is explicitly seen as the transition graph of a LLBM, as a Cayley-type graph or as a transduction graph, i.e. when the naming of its vertices is fixed, considering context-sensitive sets of final vertices does not increase the accepted class of languages.

4.3 Closure properties

Linearly bounded graphs enjoy several good properties, which will be especially important when comparing them to other classes of graphs related to LBMs or context-sensitive languages (see Section 5).

Proposition 8 *The class of linearly bounded graphs is closed under restriction to reachable vertices from any vertex and under restriction to a context-sensitive set of vertices.*

Proof We first prove the closure under restriction to a context-sensitive set of vertices. Let G be a linearly bounded graph defined by a family of $(T_a)_{a\in\Sigma}$ of 1-incremental context-sensitive transduction. By Prop. 5, the transduction $T_a \cap \{(u,v) \mid u \in F, v \in F \text{ and} |v| \leq |u|+1\}$ is 1-incremental context-sensitive. It follows that $G|_F$ is a linearly bounded graph.

Let us now prove the closure of linearly bounded graphs under restriction to reachable vertices. Let G be a linearly bounded graph given by a family of 1-incremental context-sensitive transductions $(T_a)_{a\in\Sigma}$ and u_0 a vertex in V_G. Each transduction T_a for $a \in \Sigma$ is accepted by a LBM M_a. Consider the graph G' obtained by restricting G to its set of vertices reachable from u_0.

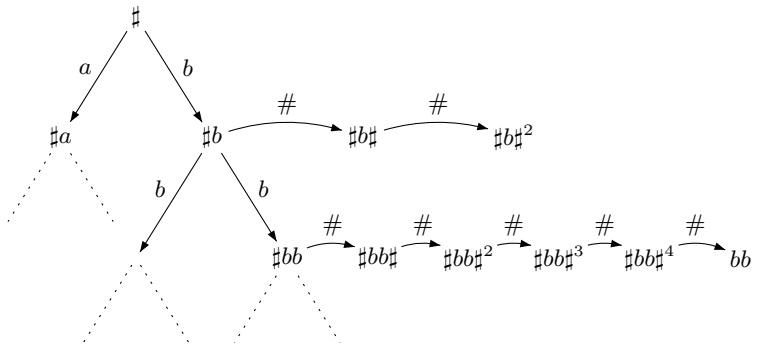

Fig. 4 The graph G for some L containing bb but not b.

Note that this set is in general not context-sensitive, as illustrated by the following example.

Example 4 Consider a language $L \subseteq \{a, b\}^*$ that is recognizable in exponential space but not in linear space (i.e. $L \in \text{NSPACE}[2^n] \setminus \text{NSPACE}[n]$). The existence of such a language is guaranteed by Thm. 2. Consider the graph G labeled by $\{a, b, \sharp\}$ and defined by:

- $\sharp u \xrightarrow[G]{x} \sharp ux$ for all $u \in \{a, b\}^*$ and $x \in \{a, b\}$,
- $\sharp u \sharp^n \xrightarrow[G]{\sharp} \sharp u \sharp^{n+1}$ for all $u \in \{a, b\}^*$ and $n + 1 \leq 2^{|u|}$,
- and $\sharp u \sharp^{2^{|u|}} \xrightarrow[G]{\sharp} u$ for all $u \in L$.

The edge relations of this graph are 1-incremental context-sensitive transductions, hence G is linearly bounded. Figure 4 illustrates the construction of this graph. The set of vertices reachable from $\sharp$ restricted to $\{a, b\}^*$ is precisely the language L which by definition is not context-sensitive.

Therefore, we need to adopt a different naming of the vertices of G'. We construct a graph H isomorphic to G' which is defined by a family of 1-incremental context-sensitive transductions. For any pair of vertices u and $v \in V_G$, we write $u \Rightarrow_n v$ if there exists a path from u to v in G using only vertices of size less than or equal to n. Note that since u and v are considered as part of any path between u and v, then necessarily $n \geq \max(|u|, |v|)$. The set of vertices of H is $V_H = \{u\Box^n \mid u \in \Gamma^*, u_0 \Rightarrow_{|u|+n} u \text{ and } u_0 \not\Rightarrow_{|u|+n-1} u\}$. Note that for all $u \in V_{G'}$, there is a unique n such that $u\Box^n \in V_H$, and that $u \in V_G \setminus V_{G'} \implies u\Box^n \notin V_H$ for all n. For all $a \in \Sigma$, we take $u\Box^n \xrightarrow[H]{a} v\Box^m$ iff $u\Box^n \in V_H$, $v\Box^m \in V_H$ and $u \xrightarrow[G]{a} v$.

The graph H is isomorphic to G'. It is easy to see that the mapping $\rho \in V_{G'} \mapsto V_H$ associating to every $u \in V_{G'}$ the unique $u\Box^n \in V_H$ is a bijection and a graph morphism from G' to H. We first prove that V_H is context-sensitive and that for all $a \in \Sigma$, $\xrightarrow[H]{a}$ is a 1-incremental context-sensitive transduction. It will then follow that H and G' are linearly bounded.

1. Consider the language L^+ equal to $\{u\Box^n \mid u_0 \Rightarrow_{|u|+n} u,\ n \in \mathbb{N}\}$ and the language L^- equal to $\{u\Box^n \mid u_0 \Rightarrow_{|u|+n-1} u,\ n \in \mathbb{N}\}$, it is easy to check that $V_H = L^+ \cap \overline{L^-}$. If we prove that L^+ and L^- are context-sensitive languages, it follows by Thm. 1 that V_H is a context-sensitive language.
   We construct a LBM M accepting L^+. When starting with $u\Box^n$, M guesses a path from u_0 to u with vertices of length at most $|u| + n$. It starts with u_0 and guesses a word u_1 of length at most $|u| + n$ then simulates one of the M_a's to check that $u_0 \xrightarrow[G]{a} u_1$. Finally, M replaces u_0 by u_1 and iterates the process until u_i is equal to u. This can be done using at most $3(|u| + n)$ cells. A similar construction allows to recognize L^-.
2. For all $a \in \Sigma$, $u\Box^n \xrightarrow[H]{a} v\Box^m$ implies that $|v| + m \leq |u| + n + 1$, because $u_0 \Rightarrow_{|u|+n} u$, $u \xrightarrow[G]{a} v$ and $\xrightarrow[G]{a}$ is 1-incremental. Therefore, $\xrightarrow[H]{a}$ is 1-incremental.
   The language $L_a = \{u\Box^n\sharp v\Box^m \mid u \xrightarrow[H]{a} v\}$ is accepted by a LBM that checks that $u\Box^n$ and $v\Box^m$ belong to V_H (by simulating a machine accepting V_H) and that $u \xrightarrow[G]{a} v$. Hence $\xrightarrow[H]{a}$ is a 1-incremental context-sensitive transduction. □

Example 5 This last construction applied to the graph of Ex. 4 gives the graph H defined by:

- $\sharp u \xrightarrow[G]{x} \sharp ux$ for all $u \in \{a, b\}^*$ and $x \in \{a, b\}$,
- $\sharp u\sharp^n \xrightarrow[G]{\sharp} \sharp u\sharp^{n+1}$ for all $u \in \{a, b\}^*$ and $n + 1 \leq 2^{|u|}$,
- and $\sharp u\sharp^{2^{|u|}} \xrightarrow[G]{\sharp} u\Box^{2^{|u|}+1}$ for all $u \in L$.

Since all rational languages are context-sensitive, linearly bounded graphs are also closed under restriction to a rational set of vertices. This shows that it is not necessary to allow arbitrary rational restrictions in the definition of transition graphs of linearly bounded machines, since such a restriction can be directly applied to the set of external configurations of a machine.

Finally, we can extend the result of [16] to the class of linearly bounded graphs, and show that they are closed under synchronized products [1]. The synchronized product of two graphs G and G' with labels in Σ and Σ' with respect to a set of constraints $C \subseteq \Sigma \times \Sigma'$ is the graph

$$G \otimes G' = \{(u, v) \xrightarrow{(a,b)} (u', v') \mid u \xrightarrow[G]{a} u' \wedge v \xrightarrow[G]{b} v' \wedge (a, b) \in C\}.$$

Proposition 9 *The class of linearly bounded graphs is closed under synchronized products, up to isomorphism.*

Proof It is straightforward from this definition that if G and G' are both linearly bounded, defined for instance as context-sensitive transduction graphs, the binary edge relations of their synchronized product are also incremental context-sensitive transductions. □

Fig. 5 Configuration graph and deterministic transition graph of a non-deterministic labeled LBM

4.4 Deterministic linearly bounded graphs

We now consider the relations between the determinism of linearly bounded graphs and the determinism of the linearly bounded machines defining them. As a first remark, it is straightforward to notice that there exist non-deterministic labeled LBMs whose transition graphs are deterministic. More precisely, any machine in which, from any given configuration, at most one external configuration is reachable by a partial run labeled by ε^*, has a deterministic transition graph. Figure 5 illustrates this fact.

In fact, we can show that any LLBM can be transformed (while preserving the accepted language) in order to verify this property. Consequently, as expressed by the following proposition, all context-sensitive languages can be accepted by a deterministic linearly bounded graph.

Proposition 10 *For every context-sensitive language L, there exists a* deterministic *linearly bounded graph G, a vertex i and a rational set of vertices F of G such that $L = L(G, \{i\}, F)$. Moreover, G can be chosen to be a tree.*

Proof Let $L \subseteq \Sigma^*$ be a context-sensitive language, we build a linearly bounded graph G labeled by Σ whose vertices are words of the form Aw or Rw, with $w \in \Sigma^*$ and $A, R \notin \Sigma$, and whose edges are defined by the following 1-incremental context-sensitive transductions:

$$Au \xrightarrow{a} Aua, \quad Rv \xrightarrow{a} Ava \qquad \text{for all } u \in L, v \notin L \text{ and } ua, va \in L,$$
$$Au \xrightarrow{a} Rua, \quad Rv \xrightarrow{a} Rva \qquad \text{for all } u \in L, v \notin L \text{ and } ua, va \notin L.$$

The language of this deterministic graph between vertex X where X is equal to A if $\varepsilon \in L$ or equal to R if $\varepsilon \notin L$ and the rational set $A\Gamma^*$ is precisely L. Moreover, G is a tree isomorphic to the complete infinite Σ-labeled tree. □

Remark 5 The previous result does not stand for a finite set of final vertices even if we only consider deterministic context-sensitive languages. Consider for example, the language $L = \{(a^n b)^* | n \in \mathbb{N}\}$. Suppose that there exists a deterministic graph G such that $L = L(G, i, F)$ for some finite set F, then there would exists $m \neq n$, such that $i \xrightarrow[G]{a^n b} f$ and $i \xrightarrow[G]{a^m b} f$. As $f \xrightarrow[G]{a^m b} f'$ for some $f' \in F$, it would follow that $i \xrightarrow[G]{a^n b a^m b} f'$.

Fig. 6 Configuration graph and deterministic transition graph of a non-deterministic terminating labeled LBM

Of course, we cannot conclude from this that the languages of deterministic linearly bounded graphs are the deterministic context-sensitive languages, which would amount to answering the general question raised by Kuroda [17] whether deterministic and non-deterministic languages coincide. However, if we only consider terminating linearly bounded machines (which have no infinite run on any given input word) the class of transition graphs we obtain faithfully illustrates the determinism of the languages.

Remark 6 Even for terminating LLBMs, the determinism of a transition graph does not imply that the machine defining it is deterministic. Figure 6 illustrates this fact. The idea of the following proof is to show that any terminating LLBM whose transition graph is deterministic can be determinized without changing the structure of the graph.

Proposition 11 *The languages of deterministic transition graphs of terminating linearly bounded machines (between an initial vertex and a rational set of final vertices) are the deterministic context-sensitive languages.*

Proof Let L be a deterministic context-sensitive language. By Prop. 2, there exists a deterministic terminating LLBM M accepting L. By definition, its transition graph G_M is deterministic, and it accepts L between the vertex corresponding to the initial configuration and those corresponding to the final configurations of M.

Conversely, let G_M be the deterministic transition graph of a terminating LLBM $M = (\Gamma, \Sigma, [,], Q, q_0, F, \delta)$, we construct a deterministic LLBM N whose transition graph G_N is equal to G_M. The machine N is obtained from M by keeping exactly one rule in δ with a given left-hand side and label. The machine N is equal to $(\Gamma, \Sigma, [,], Q, q_0, F, \delta')$ where δ' is a subset of δ:

- if $pA \xrightarrow{x} qU \in \delta$ then there exists $pA \xrightarrow{x} q'U' \in \delta'$,
- if $pA \xrightarrow{x} qU \in \delta'$ then for all $pA \xrightarrow{x} q'U' \in \delta'$, $q = q'$ and $U = U'$.

By construction, N is deterministic. Moreover, the set of external configurations of N is equal to the set of external configurations of M. It remains to prove that for any two external configurations c and c', $c \xrightarrow[G_M]{a} c'$ if and only if $c \xrightarrow[G_N]{a} c'$.

If $c \xrightarrow[G_M]{a} c'$, there exists a path between c and c' labeled by $a\varepsilon^*$ in the configuration graph C_M of M. As G_M is deterministic and terminating, any maximal path in C_M labeled by $a\varepsilon^*$ ends in c'. This property still holds for the configuration graphs of C_N. Suppose by contradiction that there exists a maximal path in C_N labeled by $a\varepsilon^*$ starting from c and ending in $c'' \neq c'$. This implies the c'' has out-going ϵ-edges in C_M and not in C_N which is impossible by construction of N.

Conversely if $c \xrightarrow[G_N]{a} c'$ then, by construction of N, we have $c \xrightarrow[G_M]{a} c'$. Hence, it follows that $G_M = G_N$. □

Remark 7 Other equivalent definitions of deterministic transition graphs of terminating labeled LBMs can be given: they coincide with the Cayley-type graphs of length-decreasing rewriting systems with unique normal forms[3], and to the graphs of deterministic terminating incremental context-sensitive transductions. The intuitive reason behind this equivalence is that, in every such case, for each label the corresponding edge relation of the graph is a partial function from vertices to vertices.

Remark 8 One can not use the previous constructions to determinize any LLBM. Indeed, the construction from Prop. 10 produces non terminating machines in general, and the construction from Prop. 1 does not preserve the structure of a machine's transition graph.

4.5 Logical properties

To conclude this section on properties of linearly bounded graphs, we investigate the decidability of the first-order theory of configuration graphs of LLBMs and linearly bounded graphs. Due to the high expressive power of the model considered, only local properties expressed in first-order logic can be checked on LLBM configuration graphs.

Proposition 12 *Configuration graphs of linearly bounded Turing machines have a decidable first-order theory.*

Proof This is a direct consequence of the fact that configuration graphs of LLBMs are synchronized rational graphs, which have a decidable first-order theory [2,15]. □

However, as remarked by [16], there exists no algorithm which, given a LLBM M and a first-order sentence ϕ, decides whether the transition graph of M satisfies ϕ. This statement can be strengthened to the following proposition.

Proposition 13 *There exists a linearly bounded graph with an undecidable first-order theory.*

[3] A rewriting system has unique normal forms if any word can be derived into at most one normal form.

Proof Consider a fixed enumeration $(M_n)_{n\in\mathbb{N}}$ of all (unlabeled) Turing machines, and a labeled Turing machine M whose language is the set of all words of the form $\#^n$ such that machine M_n halts on the empty input. Using only ε-transitions, M guesses a number n and writes $\#^n$ on its tape, then simulates machine number n on the empty input. If the machine halts, then M reads word $\#^n$ and stops. All accepting runs of M are thus labeled by $\varepsilon^*\#^n$ for some n.

When replacing ε by an observable symbol τ, the configuration graph C of M is a linearly bounded graph. Let C' be its restriction to vertices reachable from the initial configuration of M, the graph C' is still linearly bounded by Prop. 8. For all n, the formula ϕ_n expressing the existence of a path labeled by $\tau\#^n$ ending in a vertex with no successor is satisfied in C' if and only if machine M_n halts on input ε. The set of all such satisfiable formulas is not recursive. Hence the first-order theory of C' is undecidable (not even recursively enumerable). □

5 Comparison with rational graphs

We will now give some remarks about the comparison between linearly bounded graphs and several different sub-classes of rational graphs. First note that since linearly bounded graphs have by definition a finite degree, it is therefore only relevant to consider rational graphs of finite degree. However, even under this structural restriction, rational and linearly-bounded graphs are incomparable, due to the incompatibility in the growth rate of their vertices' degrees.

A first observation is that in a rational graph the out-degree at distance n from any vertex can be c^{c^n} for some constant c, whereas in a linearly bounded graph it is at most c^n.

Lemma 1 *For any linearly bounded graph G and any vertex x, there exists $c \in \mathbb{N}$ such that the out-degree of G at distance $n > 0$ of x is at most c^n.*

Proof Let $(T_a)_{a\in\Sigma}$ be a set of incremental context-sensitive transductions describing G and $k_a \in \mathbb{N}$ such that T_a is a k_a-incremental context-sensitive transduction. We take k to be the maximum of $\{k_a \mid a \in \Sigma\} \cup \{|x|\}$. At distance $n > 0$ of x, a vertex has length at most $k(n+1)$ and hence the out-degree is bounded by the number of vertices of length at most $k(n+2)$ which is less than $|\Gamma|^{k(n+2)+1}$. Hence, there exists $c \in \mathbb{N}$ such that the out-degree is bounded by c^n. □

Figure 7 shows a rational graph whose vertices at distance n from the root A have out-degree $2^{2^{n+1}}$. This graph is thus not linearly bounded.

Conversely, in a rational graph of finite degree, the in-degree at distance n from any vertex is at most c^{c^n} for some $c \in \mathbb{N}$, in a linearly bounded graph it can be as large as $f(n)$ for any mapping f from $\mathbb{N}$ to $\mathbb{N}$ recognizable in linear space (i.e. such that the language $\{0^n 1^{f(n)} \mid n \in \mathbb{N}\}$ is context-sensitive).

Lemma 2 *For any mapping $f : \mathbb{N} \mapsto \mathbb{N}$ recognizable in linear space, there exists a linearly bounded graph G with a vertex x such that the in-degree at distance $n > 0$ of x is $f(n)$.*

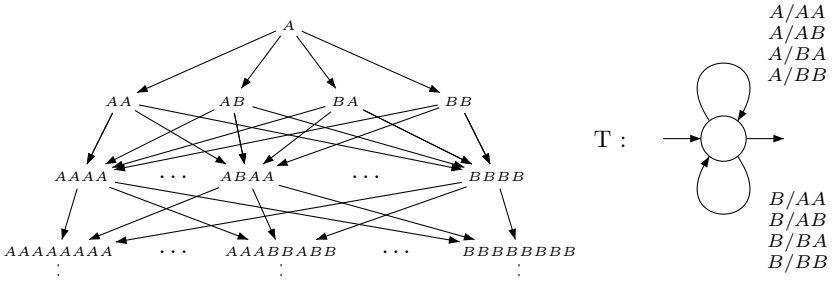

Fig. 7 A finite degree rational graph (together with its transducer) which is not isomorphic to any linearly bounded graph.

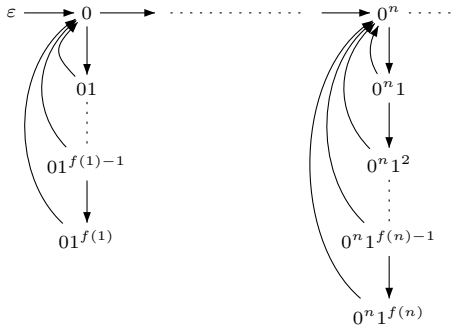

Fig. 8 A linearly bounded graph which is not isomorphic to any rational graph.

Proof Let f be a mapping from $\mathbb{N}$ to $\mathbb{N}$ recognizable in linear space, and let G_f be the linearly bounded graph defined using the following incremental context-sensitive transduction:

$$T = \{(u, u0) \mid u \in 0^*\} \cup \{(u, u1) \mid u \in 0^n 1^m,\ m < f(n)\} \cup \{(uv, u) \mid u \in 0^*, v \in 1^* \text{ and } |v| \leq f(|u|)\}.$$

Note that, since context-sensitive languages are closed under complement, it is not difficult to see that the set $\{0^n 1^m \mid m < f(n)\}$ is context-sensitive for every f recognizable in linear space. Figure 8 illustrates the construction of G_f. The in-degree of vertex 0^n at distance n from the root ε is equal to $f(n)$, for any mapping f recognizable in linear space. □

An instance of such a mapping is $f : n \mapsto 2^{2^{2^n}}$, which is more than the in-degree at distance n of a vertex in any rational graph of finite degree. From these two observations, we easily get:

Proposition 14 *The classes of finite degree rational graphs and of linearly bounded graphs are incomparable.*

Since finite-degree rational graphs and linearly bounded graphs are incomparable, we investigate more restricted sub-classes of rational graphs. For synchronized graphs of finite out-degree, we have the following result:

Proposition 15 *The synchronized graphs of finite out-degree form a strict sub-class of linearly bounded graphs.*

Proof By Prop. 4, the synchronized relations of finite image are k-incremental. Hence, the synchronized graphs of finite out-degree are linearly bounded graphs. The strictness can be deduced from the fact that synchronized graphs are not closed under restriction to reachable vertices from a given vertex (cf Prop. 8), or that synchronized graphs have decidable first-order theories (cf Prop. 13). □

By restricting the out-degree even further, we can establish the following comparison:

Theorem 5 *The rational graphs of bounded out-degree form a sub-class of linearly bounded graphs (up to isomorphism).*

Proof The inclusion result is based on a uniformization result for rational relations of bounded image due to Weber. A relation is said to be k-valued if for all $w \in \mathrm{Dom}(R)$, we have $|\{x \mid (w,x) \in R\}| \leq k$. In [28], it is proved that for any k-valued rational relation R, there exist k rational functions[4] $F_1, \dots F_k$ such that $R = \bigcup_{i\in[1,k]} F_i$.

Let $G = (T_a)_{a\in\Sigma}$ be a rational graph over Γ whose out-degree is bounded by k. Each relation T_a is therefore k-valued and hence there exist k rational functions $F_{a_1}, \dots F_{a_k}$ such that $T_a = \bigcup_{i\in[1,k]} F_{a_i}$. We write $X = \{a_i \mid a \in \Sigma \text{ and } i \in [1,k]\}$.

We will represent each vertex x of G by a pair $(w,t) \in V_G \times X^*$ such that $x = F_{t_{|t|}}(\dots(F_{t_1}(w)))$. However, there can be several such pairs for a given vertex. We define a total order $<$ on $V_G \times X^*$ and associate to x the smallest such pair. First, let $<_\Gamma$ and $<_X$ be two arbitrary total orders over Γ and X respectively and let $\prec_\Gamma$ and $\prec_X$ be the lexicographic orders induced by $<_\Gamma$ and $<_X$ respectively.

$$\begin{array}{lll}(m,t) < (n,r) & \text{iff} & |m|+|t| < |n|+|r| \\ & & \text{or } |m|+|t| = |n|+|r| \text{ and } m \prec_\Gamma n \\ & & \text{or } |m|+|t| = |n|+|r|, m = n \text{ and } t \prec_X r\end{array}$$

It is straightforward to prove that $<$ is a total order on $V_G \times X^*$. We now prove that the function N associating to any pair (w,t) the smallest pair (m,r) such that $F_t(w) = F_r(m)$ is a 0-incremental context-sensitive transduction.

There are two important points in this proof. The first one is to be able to check in linear space that for any two given pairs (m,r) and (w,t), $F_r(m) = F_t(w)$. In other terms, we want to prove that the language $L = \{m\sharp r\$t\sharp w \mid F_r(m) = F_t(w)\}$ is context-sensitive.
Let $\bar{X}$ be a finite alphabet disjoint from but in bijection with X (for all $x \in X$, we write $\bar{x}$ for the corresponding symbol in $\bar{X}$) , we consider the rational graph $\bar{G}$ defined by the family of transducers $(F_a)_{a\in X} \cup (\bar{F}_{\bar{a}})_{a\in X}$ where for all $\bar{a} \in \bar{X}$, $\bar{F}_{\bar{a}} = F_a^{-1}$. It is easy to check that for all $m, w \in \Gamma^*$

[4] A rational relation which maps exactly one image to each element of its domain is called a rational function.

and all $r, t \in X^*$, there exists a path $m \xrightarrow[\bar{G}]{r\bar{t}} w$ iff $F_r(m) = F_t(w)$ where $\bar{t} = \bar{t}_{|t|} \dots \bar{t}_1$. By Cor. 1, the language $\{i\sharp w\sharp f \mid w \in L(\bar{G}, i, f)\} \cap \Gamma^*\sharp X^*\bar{X}^*\sharp\Gamma^*$ is a context-sensitive language. It follows that L is also context-sensitive.

The second point is to show that the language $L' = \{m\sharp r\$t\sharp w \mid N((w,t)) = (m,r)\}$ is itself context-sensitive, which implies that N is indeed a context-sensitive transduction. Given (w,t) and (m,r), let $(m_1,r_1), \dots, (m_k,r_k)$ be an enumeration of all pairs smaller than (m,r) with respect to $<$. A linearly bounded machine accepting L' successively tests for all $i \in [1,k]$ whether $m_i\sharp r_i\$t\sharp w \notin L$, which is possible since L is context-sensitive and context-sensitive languages are closed under complement. If any of these k tests fails, then the whole computation fails. Otherwise, it only remains to check that $m_i\sharp r_i\$t\sharp w \in L$.

N is 0-incremental since by definition of the total order $<$, $(m,r) < (w,t) \implies |m| + |r| \leq |w| + |t|$, and all steps of the construction of the machine accepting L' can be done in linear space.

We can now define the linearly bounded graph G' with vertices in $\Gamma^*\#X^*$ using the set of transductions $(T_a)_{a\in\Sigma}$ with:

$$T_a = \bigcup_{i\in[1,k]} \{(w\#x, w'\#y) \mid N((w,x)) = (w,x) \text{ and } N(w, xa_i) = (w', y)\}.$$

As N is 0-incremental, T_a is 1-incremental. The mapping ρ from V_G to $V_G \times X^*$ associating to a vertex $x \in V_G$ the smallest (w,t) such that $x = F_t(w)$ is a bijection from V_G to $V_{G'}$, which induces an isomorphism from G to G'. Hence, G is a linearly bounded graph, which concludes the proof of inclusion. □

Example 6 Let Γ be a singleton alphabet, and consider the pair of transductions $T_a = \{(n, 2n) | n \geq 1\}$ and $T_b = \{(n, n-3) | n \geq 4\}$ on the domain of words over Γ seen as unary-coded positive integers. Let us apply the construction from the previous inclusion proof to the bounded-degree rational graph G defined by T_a and T_b. Here, $<_\Gamma$ is trivial, and we take $a <_X b$. The graph G and the linearly bounded graph G' obtained by applying the construction are shown on Fig. 9. Note for instance how vertex 13 in G is represented by vertex $(2, a^3b)$ in G' because $(2, a^3b)$ is the smallest pair (u,v) such that $u \xrightarrow[G]{v} 13$.

In fact, we can prove that the previous inclusion is strict.

Theorem 6 *There exists a linearly bounded graph of bounded degree which is not a rational graph.*

Proof We now prove that there exists a linearly bounded graph of bounded degree which is not isomorphic to any bounded degree rational graph. Let us first establish that linearly bounded graphs of bounded degree are not closed under edge reversal. Let $L \subseteq \{0,1\}^+$ be a language in NSPACE$[2^n] \setminus$ NSPACE$[n]$ (cf Thm. 2), we define a linearly bounded graph G with vertices

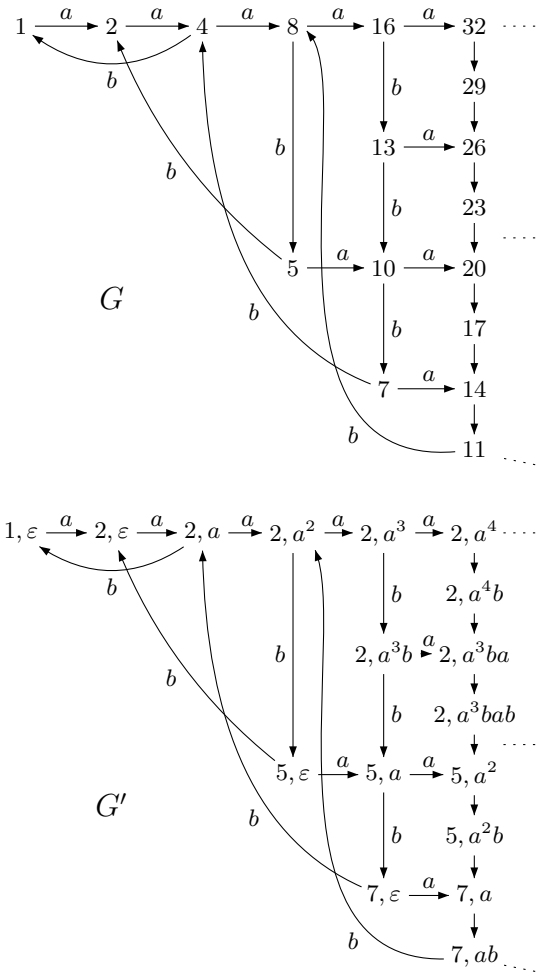

Fig. 9 Bounded-degree rational graph G and isomorphic lin. bounded graph G'.

in $0\{0,1\}^*\#^* \cup \bar{0}\{\bar{0},\bar{1}\}^*$ and edges defined by:

$$\xrightarrow{x} = \{(w, wx) \mid w \in 0\{0,1\}^*\} \cup \{(w, w\bar{x}) \mid w \in \bar{0}\{\bar{0},\bar{1}\}^*\} \qquad \text{for } x \in \{0,1\}$$

$$\xrightarrow{\#} = \{(w, w\#) \mid w = uv, u \in 0\{0,1\}^*, v \in \#^* \text{ and } |v| < 2^{|u|}\}$$
$$\cup \{(w, \bar{u}) \mid w = uv, u \in 0\{0,1\}^*, v \in \#^*, |v| = 2^{|u|} \text{ and } u \in L\}$$

The relations $\xrightarrow{0}$, $\xrightarrow{1}$ and $\xrightarrow{\#}$ are incremental context-sensitive transductions. In fact, as L belongs to NSPACE[2^n], the language $\{w\#^{2^{|w|}} \mid w \in L\}$ belongs to NSPACE[n]. The construction of G_L is illustrated by Figure 10.

Suppose that linearly bounded graphs of bounded degree were closed under edge reversal. It would follow that H obtained from G by reversing

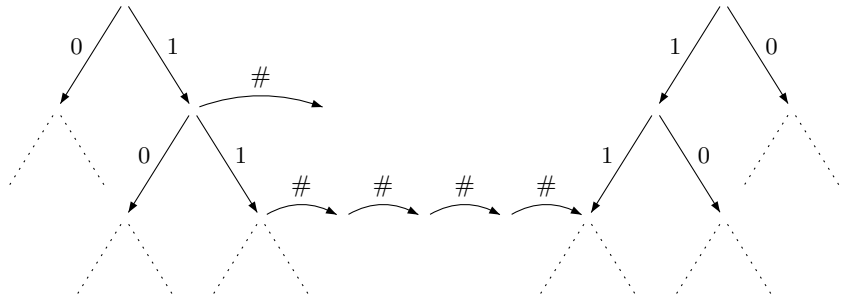

Fig. 10 The graph G for some L containing 11.

the edges labeled by # is also a linearly bounded graph. As H is linearly bounded, we can assume that $\xrightarrow[H]{0}$, $\xrightarrow[H]{1}$ and $\xrightarrow[H]{\#}$ are incremental context-sensitive transductions. Let x be the vertex of H corresponding to $\bar{0}$. The set of vertices $F = \mathrm{Dom}(\xrightarrow[H]{\#})$ is a context-sensitive language. It follows from Proposition 7 that $L(H, \{x\}, F)$ is context-sensitive. As $L = L(H, \{x\}, F) \cap \{0,1\}^*$, L would also be context-sensitive which contradicts its definition.

As rational graphs are closed under edge reversal, it follows from Thm. 5 that rational graphs of bounded degree are strictly contained in linearly bounded graphs of bounded degree. □

It may be interesting at this point to recall that there are strong reasons to believe that the languages accepted by finite degree synchronized graphs are strictly included in context-sensitive languages (see [3]). Furthermore, all existing proofs that the rational graphs accept the context-sensitive languages break down when the out-degree is bounded. It is not clear whether rational graphs of bounded degree accept all context-sensitive languages. However, as noted in Prop. 10, it is still the case for bounded degree linearly bounded graphs, and in particular for deterministic linearly bounded graphs.

6 Conclusion

This paper gives a natural definition of a class of canonical graphs associated to the observable computations of labeled linearly bounded machines. It provides equivalent characterizations of this class as the Cayley-type graphs of length-decreasing term-rewriting systems, and as the graphs defined by a sub-class of context-sensitive transductions which can increase the length of their input by at most a constant number of letters. Although of a sensibly different nature from rational graphs, we showed that all rational graphs of bounded degree are linearly bounded graphs of bounded degree, and that this inclusion is strict. This leads us to consider a more restricted notion of infinite automata, closer to classical finite automata (as was already observed in [3]), and to propose a hierarchy of classes of infinite graphs of bounded

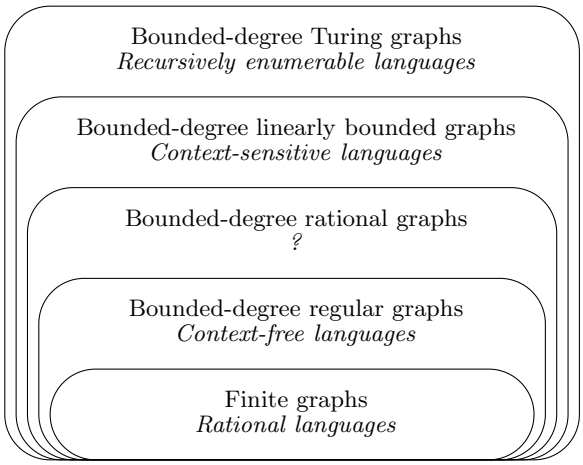

Fig. 11 A Chomsky-like hierarchy of bounded-degree infinite graphs.

degree accepting the classes of languages of the Chomsky hierarchy, in the spirit of [8] (see Fig. 11).

Finite graphs obviously have a bounded degree, they accept rational languages. The transition graphs of real-time pushdown automata, which accept all context-free languages, are the regular graphs [22], or equivalently the bounded degree HR graphs [10] and bounded degree prefix-recognizable graphs [7]. By Prop. 10, the languages of deterministic linearly bounded graphs are the context-sensitive languages. Deterministic graphs have by definition a bounded degree, so bounded degree linearly bounded graphs also accept the same class of languages. Finally, since deterministic Turing machines have bounded-degree transition graphs and accept all recursively enumerable languages, we can also restrict the class of Turing graphs to bounded degree.

References

1. Arnold, A., Nivat, M.: Comportements de processus. In: Colloque de l'Association Francaise pour la Cybernétique Économique et Théorique: Les Mathématiques de l'Informatique (AFCET '82), pp. 35–68 (1982)
2. Blumensath, A., Grädel, E.: Automatic structures. In: Proceedings of the 15th IEEE Symposium on Logic in Computer Science (LICS 2000), pp. 51–62. IEEE (2000)
3. Carayol, A., Meyer, A.: Context-sensitive languages, rational graphs and determinism (2005). Submitted
4. Caucal, D.: On infinite transition graphs having a decidable monadic theory. In: Proceedings of 23rd International Colloquium on Automata, Languages, and Programming (ICALP 1996), *Lecture Notes in Computer Science*, vol. 1099, pp. 194–205. Springer Verlag (1996)
5. Caucal, D.: On infinite transition graphs having a decidable monadic theory. Theoretical Computer Science **290**, 79–115 (2003)
6. Caucal, D.: On the transition graphs of Turing machines. Theoretical Computer Science **296**, 195–223 (2003)

7. Caucal, D., Knapik, T.: An internal presentation of regular graphs by prefix-recognizable graphs. Theoretical Computer Science **34**, 299–336 (2001)
8. Caucal, D., Knapik, T.: A Chomsky-like hierarchy of infinite graphs. In: Proceedings of the 27th International Symposium on Mathematical Foundations of Computer Science (MFCS 2002), *Lecture Notes in Computer Science*, vol. 2420, pp. 177–187. Springer Verlag (2002)
9. Chomsky, N.: On certain formal properties of grammars. Information and Control **2**, 137–167 (1959)
10. Courcelle, B.: The monadic second-order logic of graphs, II: Infinite graphs of bounded width. Mathematical System Theory **21**, 187–221 (1989)
11. Elgot, C., Mezei, J.: On relations defined by finite automata. IBM Journal of Research and Development **9**, 47–68 (1965)
12. Frougny, C., Sakarovitch, J.: Synchronized rational relations of finite and infinite words. Theoretical Computer Science **108**(1), 45–82 (1993)
13. Hopcroft, J., Ullman, J.: Introduction to Automata Theory, Languages and Computation. Addison-Wesley (1979)
14. Immerman, N.: Nondeterministic space is closed under complementation. SIAM Journal on Computing **17**(5), 935–938 (1988)
15. Khoussainov, B., Nerode, A.: Automatic presentations of structures. In: Selected papers of the 1994 International Workshop on Logic and Computational Complexity, *Lecture Notes in Computer Science*, vol. 960, pp. 367–392. Springer Verlag (1994)
16. Knapik, T., Payet, É.: Synchronized product of linear bounded machines. In: Proceedings of the 12th International Symposium on Fundamentals of Computation Theory, (FCT 1999), *Lecture Notes in Computer Science*, vol. 1684, pp. 362–373. Springer Verlag (1999)
17. Kuroda, S.: Classes of languages and linear-bounded automata. Information and Control **7**(2), 207–223 (1964)
18. Latteux, M., Simplot, D., Terlutte, A.: Iterated length-preserving rational transductions. In: Proceedings of the 23rd International Symposium on Mathematical Foundations of Computer Science (MFCS 1998), *Lecture Notes in Computer Science*, vol. 1450, pp. 286–295. Springer Verlag (1998)
19. Milner, R.: Communication and Concurrency. Prentice Hall International (1989)
20. Morvan, C.: On rational graphs. In: Proceedings of the 3rd International Conference on Foundations of Software Science and Computation Structures (FoSSaCS 2000), *Lecture Notes in Computer Science*, vol. 1784, pp. 252–266. Springer Verlag (2000)
21. Morvan, C., Stirling, C.: Rational graphs trace context-sensitive languages. In: Proceedings of the 26th International Symposium on Mathematical Foundations of Computer Science (MFCS 2001), *Lecture Notes in Computer Science*, vol. 2136, pp. 548–559. Springer Verlag (2001)
22. Muller, D.E., Schupp, P.E.: The theory of ends, pushdown automata, and second-order logic. Theoretical Computer Science **37**, 51–75 (1985)
23. Payet, É.: Thue specifications, infinite graphs and synchronized product. Ph.D. thesis, Université de la Réunion (2000)
24. Rispal, C.: The synchronized graphs trace the context-sensitive languages. In: Proceedings of the 4th International Workshop on Verification of Infinite-State Systems (INFINITY 2002), *Electronic Notes in Theoretical Computer Science*, vol. 68 (2002)
25. Stirling, C.: Decidability of bisimulation equivalence for pushdown processes. Tech. Rep. EDI-INF-RR-0005, School of Informatics, University of Edinburgh (2000)
26. Szelepcsenyi, R.: The method of forced enumeration for nondeterministic automata. Acta Informatica **26**, 279–284 (1988)
27. Thomas, W.: A short introduction to infinite automata. In: Proceedings of the 5th International Conference on Developments in Language Theory (DLT 2001), *Lecture Notes in Computer Science*, vol. 2295, pp. 130–144. Springer Verlag (2001)
28. Weber, A.: Decomposing a k-valued transducer into k unambiguous ones. Informatique Théorique et Applications **30**(5), 379–413 (1996)